John von Neumann Institute for Computing

Load Balanced Parallel Simulation of Particle-Fluid DEM-SPH Systems with Moving Boundaries

Florian Fleissner, Peter Eberhard

published in

Parallel Computing: Architectures, Algorithms and Applications ,
C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr, F. Peters (Eds.),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **38**, ISBN 978-3-9810843-4-4, pp. 37-44, 2007.
Reprinted in: Advances in Parallel Computing, Volume **15**,
ISSN 0927-5452, ISBN 978-1-58603-796-3 (IOS Press), 2008.

http://www.fz-juelich.de/nic-series/volume38

Load Balanced Parallel Simulation of Particle-Fluid DEM-SPH Systems with Moving Boundaries

Florian Fleissner and Peter Eberhard

Institute of Engineering and Computational Mechanics
University of Stuttgart, 70550 Stuttgart, Germany
E-mail: {fleissner, eberhard}@itm.uni-stuttgart.de

We propose a new pure Lagrangian method for the parallel load balanced simulation of particle-fluid systems with moving boundaries or free surfaces. Our method is completely meshless and models solid objects as well as the fluid as particles. By an Orthogonal Recursive Bisection we obtain a domain decomposition that is well suited for a controller based load balancing. This controller approach is designed for being used on clusters of workstations as it can cope with load imbalances not only as emerging from the simulation dynamics but also from competitive processes of other users. In this paper we present the most important concepts such as neighbourhood search, particle interactions, domain decomposition and controller based load balancing.

1 Introduction

Various fields of engineering require the simulation of particle-fluid systems, such as dispersions with dynamically moving boundaries or free surfaces. Modelling these systems with a mixed Eulerian-Lagrangian approach requires a complicated coupling of a grid based and a meshless method, which is especially difficult if computations are to be carried out in parallel. The consideration of boundary conditions, the distribution of the workload to the parallel nodes and the a priori prediction of the simulation domain is a difficult task. Therefore, we propose a new meshless pure Lagrangian approach, that overcomes all of the problems mentioned above.

We start in Section 2 with a description of our simulation model. Therefore, we introduce the two methods for the simulation of rigid particles and the fluid and explain how the complexity of the detection of particle interactions can be reduced by an efficient neighbourhood search. We also describe how the two particle types are coupled in terms of particle interactions. The implementation of simulation boundaries is explained in Section 3. The most important aspects of our parallel simulation approach, such as process synchronization, domain decomposition and load balancing are finally introduced in Section 4.

2 Pure Lagrangian Particle-Fluid Simulation

For our pure Lagrangian approach we employ Smoothed Particle Hydrodynamics[1] (SPH) as a particle based fluid simulation method and the Discrete Element Method (DEM)[2,3] for the simulation of solid particles. The main challenges of the implementation of a pure Lagrangian approach are the coupling of the different particle types, fluid and dry particles as well as the consideration of domain boundaries. Where for grid based fluid simulation methods boundary conditions have to be imposed on the underlying PDE, for particle

boundary conditions can be considered as penalty forces that prevent particle-particle or particle-wall penetrations. These penalty forces are simply added as extra terms to the right hand sides of Newton's or Euler's equations of motion.

2.1 Discrete Element Method

The Discrete Element Method[2,3] models particles as rigid bodies whose dynamics is described by Newton's and Euler's equations of motion. Forces resulting from particle interactions are accumulated, serving as right hand sides of the dynamic equations of motion. Typical types of interactions are e.g. contact forces, modelled by linear or non-linear contact springs, or potentials such as Lennard-Jones potentials[4] that provide attractive or repelling forces depending on the distance of two particles.

2.2 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics[1] calculates field variables such as pressure and fluid velocities only at freely moving discrete points in space. As to every point a constant mass is assigned, it can be considered as a particle. Field variables can be evaluated by evaluating the kernel functions of the particles in the close vicinity and superimposing the results. These interpolation kernel functions, such as the Gaussian kernel, are required to have a finite cutoff radius that restricts their domain. By introducing this approximation in the strong form of the Navier-Stokes equations, the space- and time-dependent PDEs can be transformed into only time-dependent ODEs that describe the motion of the individual particles, yet considering for the effects imposed by pressure and viscosity of adjacent particles.

2.3 Neighbourhood Search

Both particle approaches require the determination of adjacent, potentially interacting particle pairs. To avoid a costly $O(n^2)$ search for interacting particle pairs out of n particles, there exists a large variety of different neighbourhood search algorithms that reduce the complexity to $O(n)$. The existing approaches come with different advantages and disadvantages for different types of interacting geometries[5–9]. To allow for general particle shapes and polydisperse particle systems, we combine two methods based on axis-aligned bounding boxes (AABB)[10,11] to gain a multi-purpose method that is well suited for a hierarchically structured application, an important prerequisite for our parallelization approach[12]. All geometrical entities, including boundaries, are defined as particles surrounded by bounding boxes. An interaction between two particles can only occur if their bounding boxes overlap. A necessary condition for a bounding box overlap is an overlap of the projections of the boxes on the three spatial axes, see Fig. 1.

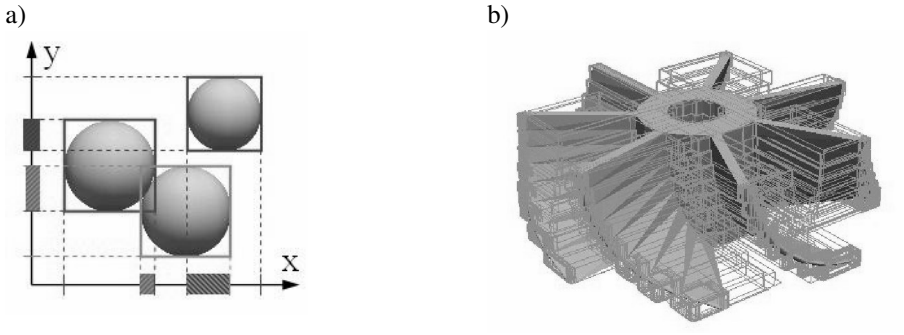

Figure 1. a) Neighbourhood search with axis aligned bounding boxes, depicted in 2D. Particle interactions only occur if the projections of two particles' bounding boxes on all three (3D) spatial axis overlap. b) Triangulated geometrical model of a turbine wheel with bounding boxes for neighbourhood search.

2.4 DEM-SPH Coupling

To couple a solid particle i and a SPH particle j, we employ Lennard-Jones potentials

$$d_{ij} = |\boldsymbol{r}_{ij}|, \tag{1}$$

$$\boldsymbol{n}_{ij} = \frac{\boldsymbol{r}_{ij}}{d_{ij}}, \tag{2}$$

$$\boldsymbol{F}_{ij}^{n} = \frac{\epsilon}{d_0}\left(\left(\frac{d_0}{d_{ij}}\right)^{7} - \left(\frac{d_0}{d_{ij}}\right)^{13}\right)\boldsymbol{n}_{ij}, \tag{3}$$

that may be interpreted as stiff nonlinear penalty forces. Their forces depend on the particle distance vector $\boldsymbol{r}_{ij}$ and a zero force distance d_0. To incorporate no-slip boundary conditions, we add the viscous tangential forces

$$\boldsymbol{v}_{ij}^{t} = \boldsymbol{v}_{ij} - (\boldsymbol{v}_{ij} \cdot \boldsymbol{n}_{ij})\,\boldsymbol{n}_{ij}, \tag{4}$$

$$\boldsymbol{F}_{ij}^{t} = -k\boldsymbol{v}_{ij}^{t} \tag{5}$$

to gain the total force

$$\boldsymbol{F}_{ij} = \boldsymbol{F}_{ij}^{n} + \boldsymbol{F}_{ij}^{t}. \tag{6}$$

A nearly incompressible fluid can thus be simulated by choosing the parameter ϵ of the Lennard-Jones potential in a way that yields a stiff repelling behaviour.

3 Simulation Boundaries

Simulations of particle-fluid systems in engineering applications often involve boundaries with complex geometry, e.g. provided as CAD data. We propose an approach that is based on a surface triangulation of the boundary geometry. There exists a large variety of open source tools for the triangulation of CAD data. All surface triangles are treated as individual particles in terms of neighbourhood search and their distribution to parallel processor nodes. Particle-triangle interactions are computed as described in[13] with Lennard-Jones

a) b)

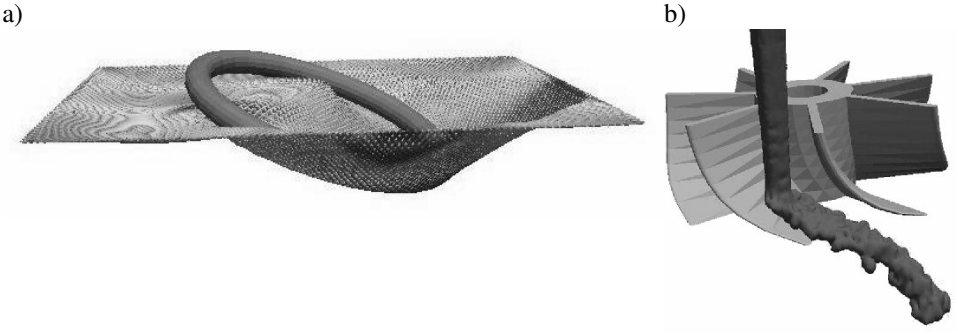

Figure 2. a) Simulation of a rigid torus falling on a membrane. The torus consists of surface triangles which are treated individually during neighbourhood search and interaction computation. b) Simulation of a turbine wheel driven by a liquid jet. The surface of the jet is reconstructed and rendered.

contact forces instead of penalty springs. Visco-elastic continua such as strings, membranes or solids are modelled by bonding particles with visco-elastic rods. For the simulation of dynamically moving rigid bodies, the triangulated surface geometry is defined relative to a moving frame of reference that possesses mass and inertia properties, such as the torus and the turbine wheel depicted in Fig. 2.

4 Parallel Simulation

For parallel simulations we apply a spatial domain decomposition in order to distribute particles to computation nodes of the parallel processor. Our approach is based on the point-to-point communication paradigm as featured by the Message Passing Interface (MPI) or the Parallel Virtual Machine (PVM). We employ a manager-worker modell where the manager process coordinates the work distribution and maintains the load balancing. Communication between the worker processes is initialized dynamically by the manager process, see Fig. 3.

4.1 Process Synchronization

Several substeps of the particle simulation loop require different communication patterns (m: manager, w: worker) and have to be partially synchronized:

- Neighbourhood search and interaction computation (w↔w),
- integration (w→m→w),
- particle inter-node migration (w→m→w, w↔w),
- post-processing data output (w→m),
- load-balancing (w→m→w).

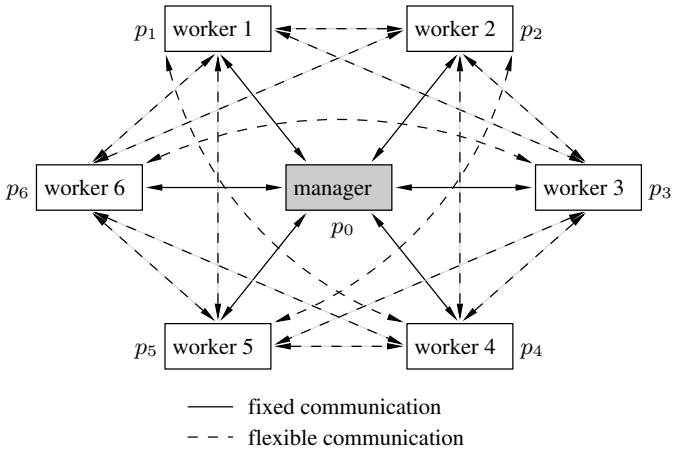

Figure 3. The communication between processes is based on a fixed-flexible communication pattern that is realized via point-to-point communication in MPI. Any communication between the worker processes that is based on the system state (flexible) is initialized by the manager process.

There are substeps which require a great amount of worker to worker communication (w↔w), such as neighbourhood search, and others that require no worker to worker communication, such as post-processing data output, which are handled by the manager process. As we apply point-to-point communication, there is no need to use semaphores to ensure the synchronization of all nodes in all substeps. The worker processes can be allowed to work on until the next synchronization point occurs.

4.2 Domain Decomposition

As a domain decomposition scheme that can be dynamically adapted, we chose Orthogonal Recursive Bisection (ORB)[14,15]. Processors are assigned to subdomains that are logically corresponding to the leaves of the binary tree that emerges from the ORB decomposition. To setup the decomposition, subdomains are consecutively divided in half along one of the three spatial axis. The requirement that the particles in the two emerging subgroups match the accumulated computational power of the nodes in the two corresponding subtrees[12] serves as a constraint for the placement of the division boundary, see Fig. 4. For the placement of the subdivision boundaries, we divide a particle cloud in two subclouds whose cardinalities match the same ratio as two given accumulated computational power values. This task is accomplished by a sampling of the particle positions on a regular grid. The grid based distribution function is used to compute normalized cumulative particle density functions for the three spatial dimensions which can in turn be evaluated to find the appropriate boundary position.

4.3 Load Balancing

A balanced distribution of the simulation work is crucial to gain optimal performance in parallel computations. Overworked nodes can slow down a parallel computation and can

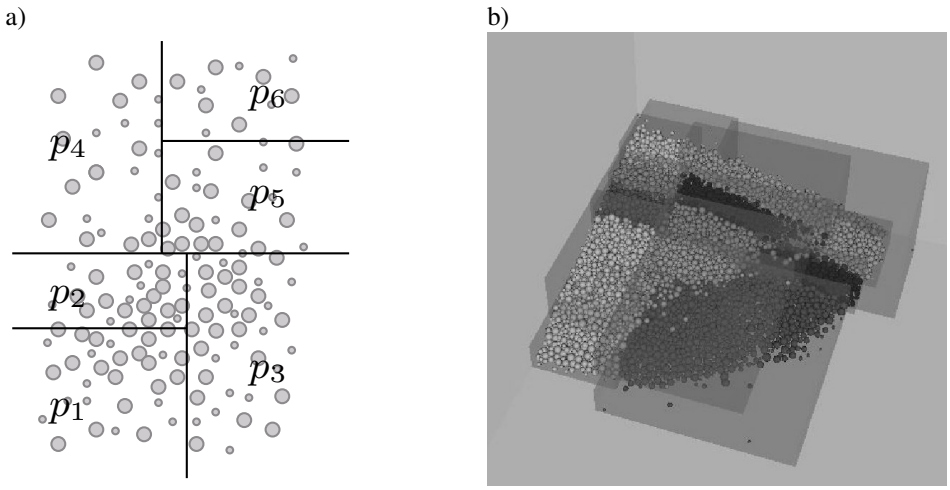

Figure 4. Orthogonal Recursive Bisection: a) The simulation domain is recursively bisected to obtain as many subdomains as nodes. b) Assignment of particles to 16 nodes (coloured) for a 3D simulation.

thus lead to execution times that are in the worst case even longer than that of equivalent sequential computations. To avoid imbalances, the workload needs to be continuously redistributed between the worker nodes. There are two main reasons for an imbalance of workload. Firstly, particle motion requires migration of particles between processors and can thus cause an imbalance of the number of particles assigned to the nodes. Secondly, other users can run competitive jobs that may influence the current performance of the nodes. The latter, however, only occurs in clusters of workstations where nodes are not exclusively reserved for a particular job.

Our load balancing approach works independent from the reason for the imbalance. The basic idea is to shift subdomains boundaries on every level of the ORB-tree. Shifting boundaries causes particles to be migrated to neighboring domains and thus leads to an improved distribution of workload. As imbalances are non deterministic, a control-approach is required. We employ a hierarchical proportional-integral (PI)-controller[16] with the differences of computation times of two twin nodes as controller input. The same approach is applied on all levels of the ORB-tree with groups of interior nodes instead of leafs. Therefore, the computation times of the nodes in a group are accumulated and the differences between the accumulated computation times of two twin groups are used as controller input[12]. See Fig. 5 for an example of the control process employing six worker nodes.

4.4 Performance

As a benchmark for the performance of our simulation approach, we adopted the collapsing block example as depicted in Fig. 4. Simulation series were performed on a cluster of Pentium 4 workstations with ten thousand, hundred thousand and one million particles. The results are depicted in Fig. 6. As expected, the scaling behaviour improves with increasing problem size. However, performance is significantly affected by communication.

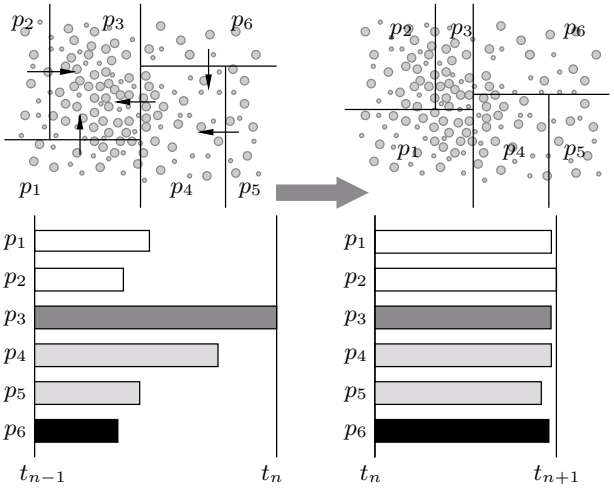

Figure 5. Controller approaches for a six node scheme. During a discrete controller step, all subdivision boundaries of the ORB decomposition are shifted. After the controller step the nodes' per step wall clock computation times have adapted and the total per time step wall time has been reduced.

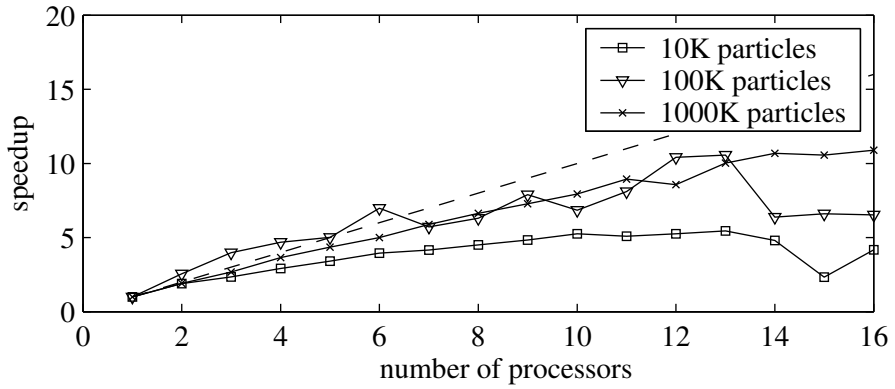

Figure 6. Parallel speedup of simulations of the collapsing particles example, see Fig. 4, for simulation series with ten thousand, hundred thousand and one million particles. The number of processors includes the manager and the worker processes.

5 Summary

We presented a new parallel method for the simulation of particle-fluid systems in a pure Lagrangian way that does not involve any grid based methods. Our method is thus well suited for applications where the simulation domain is not a priory known, such as free surface flows or flows with moving, e.g. elastic boundaries. The method combines Smoothed Particle Hydrodynamics and the Discrete Element Method, both meshless particle meth-

ods. The load balancing approach is based on Orthogonal Recursive Bisection. It is capable of maintaining load balance even on clusters of workstations with competitive jobs. Its PI-controller, controlls the amount of particles assigned to the different nodes based on the differences of their per step computation times.

References

1. J. J. Monaghan, *Smoothed particle hydrodynamics*, Annual Reviews in Astronomy and Astrophysics, **30**, 543–574, (1992).
2. P. A. Cundall, *A computer model for simulation of progressive, large-scale movements in blocky rock systems*, in: Proc. Symposium of the International Society of Rock Mechanics, Nancy, (1971).
3. P. A. Cundall and O. D. L. Strack, *A discrete numerical model for granular assemblies*, Geotechnique, **29**, 47–56, (1979).
4. L. Verlet, *Computer experiments on classical fluids*, Physics Reviews, **159** ,98–103, (1967).
5. G. v.d. Bergen, *Efficient collision detection of complex deformable models using aabb trees*, J. Graphics Tools Archive, **2**, 1–13, (1998).
6. D. James and D. Pai, *Bd-tree: Output-sensitive collision detection for reduced deformable models*, in: ACM Trans. Graphics (SIGGRAPH 2004), (2004).
7. D. S. Coming and O. G. Staadt, *Kinetic sweep and prune for multi-body continuous motions*, in: Second Workshop in Virtual Reality Interactions and Physical Simulations, (2005).
8. M. Teschner, S. Kimmerle, G. Zachmann, B. Heidelberger, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnetat-Thalmann, and W. Strasser, *Collision detection for deformable objects*, in: Eurographics State-of-the-Art Report (EG-STAR), pp. 119–139, Eurographics Association, (2004).
9. M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets and M. Gross, *Optimized spatial hashing for collision detection of deformable objects*, in: Proc. Vision, Modeling, Visualization VMV'03, pp. 47–54, (2003).
10. A. Schinner, *Fast algorithms for the simulation of polygonal particles*, Granular Matter, **2**, 35–43, (1999).
11. E. Perkins and J. R. Williams, *A fast contact detection algorithm insensitive to object sizes*, Engineering Computations, **18**, 48–61, (2001).
12. F. Fleißner and P. Eberhard, *Parallel load balanced simulation for short range interaction particle methods with hierarchical particle grouping based on orthogonal recursive bisection*, International Journal for Numerical Methods in Engineering, (2007, in press).
13. F. Fleißner, T. Gaugele, and P. Eberhard, *Applications of the discrete element method in mechanical engineering*, Multibody System Dynamics, **18**, 81–94, (2007).
14. J. Salmon, *Parallel hierarchical N-body methods*, PhD thesis, Caltech University, (1990).
15. M. S.Warren and J. K. Salmon, *A parallel hashed oct-tree n-body algorithm*, in: Proc. Supercomputing '93, pp. 12–21, (1993).
16. R. C. Dorf and R. H. Bishop, *Modern control systems*, (Prentice Hall, New Jersey, 2001).